Designing an Automated System for Medical Diagnosis

Ranjan Parekh

Abstract

This paper proposes an automated system for recognizing disease conditions of human skin in context to health informatics. Skin texture images, displaying three dermatological skin conditions, are analyzed using a texture analysis technique, based on a set of normalized symmetrical Grey Level Co-occurrence Matrices (GLCM), and features are extracted from them using automated algorithms. The features are fed to neural network classifiers for identification of the disease type. The features are considered in various combinations viz. individually, in joint 2-D and 3-D feature spaces, to find out the best recognition accuracies.

Keywords

Medical image analysis, texture recognition, grey level co-occurrence matrix (GLCM), neural networks, computer vision.

1. Introduction

In recent years, computer vision methodologies have been applied to the fields of health informatics and telemedicine for automated diagnosis of diseases. Interest in automated health diagnosis has been triggered by the huge collection of medical images generated everyday all over the world. For example, the Radiology Department of the University Hospital of Geneva alone produces more than 12,000 images a day [Muller, 2004]. Automated diagnosis measures have shown great potentials for reducing diagnostic errors and improving the accuracy and efficiency of medical diagnosis. Diagnostic errors have huge negative impact on patient care, such as an incremental cost per patient and increase in hospital stay as reported in a Harvard study [Bates, 1997]. In the United States alone, medical error results in 44,000-98,000 unnecessary deaths each year and 1,000,000 excess injuries [Weingart, 2000]. Ironically, most diagnostic errors are preventable. Research shows that diagnosis errors often occur when clinicians are inexperienced and new procedures are introduced. Further, age, complex care, urgent care, and prolonged hospital stay have been found to be correlated with diagnostic errors. Application of automated information systems in medical analysis has shown great promise in reducing such errors [Copec, 2003]

Automated diagnostic systems based on medical imaging, work by using image processing techniques to recognize and differentiate disease characteristics from digital images. Two important steps are used : (1) Visual features are extracted from the images and represented using a mathematical data model (2) the data representation is then fed to a statistical classifier like a neural network, for identification and classification of the disease.

Image features usually involve the following either individually or in various combinations : color, texture, shape. In this paper we focus on using texture as a means of identifying skin diseases from medical images using an automated procedure. Texture refers to visual patterns for describing the variation of color or grey tones over the image. The choice of the features, depend on what characteristics we are trying to identify from the image that best describes the diseases in question. In many cases computer system designers need to take help from medical professionals to isolate the best set of features most useful in describing a specific disease condition. The organization of the paper is as follows : section 2 provides an overview of the related work, section 3 outlines the proposed approaches with discussions on overview, feature computation and classification schemes, section 4 provides details of the dataset and experimental results obtained, section 5 provides an analysis of the current work vis-à-vis other related works, and section 6 provides the overall conclusion and scope for further research.

2. Related Works

Many methodologies have been proposed to analyze and recognize textures and shapes in an automated fashion. One of the first studies involved derivation of texture energy measures using a set of simple masks (vertical, horizontal, diagonal and anti-diagonal) [Wang, 1986]. Authors like Tamura [Tamura, 1978] made an attempt at defining a set of visually relevant texture features. This includes coarseness, contrast, directionality, line-likeness, regularity, roughness. Pentland [Pentland, 1984] reports a high degree of correlation between fractal dimensions and human estimates of roughness. Two-state Markov models have been used to detect texture edges characterized by changes in first order statistics [Huang, 1984]. Gabor filters have been used in several image analysis applications including texture classification and segmentation [Bovik, 1990].

Specifically, computer vision techniques involving texture analysis have been applied to health informatics to predict and characterize skin diseases. N. K. Al abbadi et al. [Alabbadi, 2008] have proposed a method for skin texture recognition using a 3 layer neural network using both skin color and texture features. In [Rubegni, 2002] authors propose a method of diagnosis of pigmented skin lesions by using a digital dermoscopy analyzer to evaluate a series of clinically atypical, flat pigmented skin lesions. Fractal parameters such as lacunarity and fractal dimensions have been used in diagnosis of skin cancers [Blackledge, 2009]. The use of Bayesian networks for skin texture recognition has been reported in [Shahreza, 2008]. A review of image analysis techniques for medical diagnosis can be found in [Muller, 2004].

3. Proposed Approach

3.1 GLCM : An Overview

This section describes how a popular texture modeling technique called Grey Level Co-occurrence Matrix (GLCM) can be used to model texture content in images. A GLCM [Haralick, 1979] indicates probability of a grey-level i occurring in the neighbourhood of grey-level j at a distance d and direction θ.

$$G=P(i,j|d,\theta) \tag{1}$$

GLCMs can be computed from texture images using different values of d and θ and these probability values create the co-occurrence matrix. Consider a 4 by 4 section **I** of an image having four grey-level intensities as shown below.

$$\mathbf{I}=\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

To compute the frequency of one grey tone in the neighbourhood of others, a 4×4 matrix is formed (since there are four distinct grey tones) and sequential numbers along the left (reference) and top (neighbour) are used to indicate them. The frequencies in which each pair (reference-neighbour) of grey-tones, occur together in **I** is now computed i.e. for a reference grey-tone i, how many times the neighbour grey-tone j occurs near it within **I**, and this constitutes the (i,j)-th element of GLCM matrix G. For simplicity's sake we consider the distance d as 1 i.e. only adjacent pixels are considered, and angle θ as 0° i.e. along the positive x-axis from left to right.

$$\mathbf{G}=\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 2 & 2 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 3 & 1 \\ 3 & 0 & 0 & 0 & 1 \end{array}$$
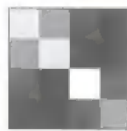
For example, 0 (reference) adjacent to 0 (neighbour) in **I** occurs 2 times (rows 1 and 2), hence we put 2 at position (0,0) of **G**, 0 adjacent to 1 occurs 2 times (rows 1 and 2) hence (0,1) contains 2, 0 adjacent to 2 occurs 1 time (row 3) hence (0,2) contains 1 and so on. This procedure is repeated for all pairs of intensities.
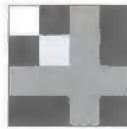
If we had moved along the −ve x-axis, i.e. we had looked from right to left, then the matrix formed would have been the transpose matrix $\mathbf{G}^T$. To make the matrix independent of this factor, the transpose is added to the original to make it symmetrical viz. $\mathbf{S} = \mathbf{G} + \mathbf{G}^T$:

$$\mathbf{G}+\mathbf{G}^T=\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}+\begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}=\begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}=\mathbf{S}$$

The symmetrical GLCM is finally normalized by dividing each element by the sum of all elements to form $\mathbf{S_0}$. The '0' in the subscript indicates angle $\theta = 0°$. Directional GLCMs can also be computed along three other directions: vertical ($\theta = 90°$), right diagonal ($\theta = 45°$) and left diagonal ($\theta = 135°$) generating matrices $\mathbf{S_{45}}, \mathbf{S_{90}}$, and $\mathbf{S_{135}}$:

$$\mathbf{S_0}=\frac{1}{24}\begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

$$\mathbf{S_{45}}=\frac{1}{18}\begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{S_{90}}=\frac{1}{24}\begin{bmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

$$\mathbf{S_{135}}=\frac{1}{18}\begin{bmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

3.2 GLCM based Features

An 8-bit grayscale image typically contains 256 different grey tones which generates GLCMs having

256 by 256 elements. Since it is inconvenient to deal with such large matrices, a set of scalar features are usually derived from directional normalized symmetrical GLCMs and used for texture characterization viz. GLCM Contrast (©), GLCM Homogeneity (H), GLCM Mean (M), GLCM Variance (V) and GLCM Energy (N) as defined in Eq. (2). Here $S_{i,j}$ represents the element (i,j) of a normalized symmetrical GLCM, and k the number of grey levels.

$$C = \sum_{i=1}^{k} \sum_{j=1}^{k} S_{i,j} (i - j)^2$$

$$H = \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{S_{i,j}}{1 + (i - j)^2}$$

$$M = M_i = \sum_{i=1}^{k} \sum_{j=1}^{k} i S_{i,j} = M_j = \sum_{i=1}^{k} \sum_{j=1}^{k} j S_{i,j} \qquad (2)$$

$$V = \sum_{i=1}^{k} \sum_{j=1}^{k} S_{i,j} (i - M_i)^2 = \sum_{i=1}^{k} \sum_{j=1}^{k}$$

$$N = \sqrt{\sum_{i=1}^{k} \sum_{j=1}^{k} S^2_{i,j}}$$

3.3 GLCM based Classification

A texture class i consists of a set of n member images: $T_i\{t_1, t_2, ... t_n\}_i$. For each member image, four directional symmetrical normalized GLCMs are computed as indicated below :

$$\{(t_0, t_{45}, t_{90}, t_{135})\}_1, ..., (t_0, t_{45}, t_{90}, t_{135})_n\}_i$$

For each directional GLCM, features in Eq. (2) are computed. Each feature is averaged over the four directional GLCMs, for each member image viz.

$$\{(\bar{t}_x)_1, ..., (\bar{t}_x)_n\}_i$$

where, $t_{\bar{x}} = \frac{t_{x,0} + t_{x,45} + t_{x,90} + t_{x,135}}{4}$ and $X \in \{C, H, M, N, V\}$. A texture class is characterized by the collection of its feature values obtained during a training phase. A test image S_j with its computed average features $(S_{\bar{x}})_j$ is said to belong to a specific texture class if the probability of its feature values being a member of that training class is maximum. To compute class probability neural network classifiers (multi-layer perceptron : MLP) using feed-forward back-propagation architectures are employed in this work.

3.4 Neural Networks : An Overview

Artificial neural networks are a set of algorithms meant to simulate workings of human nerve cells or neurons. A neuron receives input stimuli from a number of sources using an electro-chemical process and produces a response ('fires') when concentration of electrical charges exceeds a certain threshold. An artificial neural unit is also visualized as a structure having n inputs, and each input channel i can carry a signal xi. The neural unit can produce an output o when the sum of the input signals exceeds a certain pre-defined threshold θ i.e. when $x_1 + x_2 + ... + x_n \geq \theta$. See Fig.1.

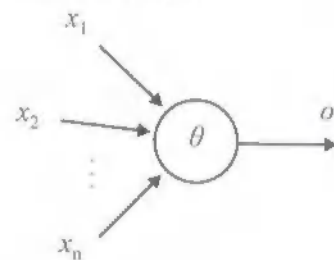Fig. 1. Artificial neuron unit

One of the earliest neurons is the McCulloh- Pitts neuron where the inputs and outputs are considered as binary values. Fig. 2 shows simple implementations of 'AND' and 'OR' logical gates using McCulloh-Pitts neurons having thresholds of 2 and 1 respectively.
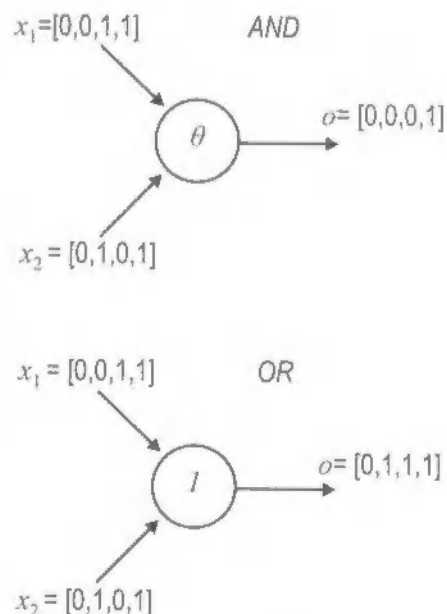

Fig. 2. Using McCulloh-Pitts neurons to implement AND and OR logical gates

The perceptron was proposed as a more generalized model of the McCulloh-Pitts neuron and is nowadays extensively used in classification problems in various domains. The essential difference is the presence of numerical weights associated with input lines. See Fig. 3. This imparts the perceptron the capability to *learn* by adapting the weights to suit a particular classification problem. The input signals and weights are now no longer restricted to binary values but can take on any real value. The input signals are designated by an input vector $X=\{x_1, x_2,...,x_n\}$ and the weights by a weight vector $W=\{w_1, w_2,...,w_n\}$. The perceptron also has a bias line connected to a bias signal (B) kept permanently 1 and an associated bias weight (b). The net input N to the perceptron is given by :

$$N=b+W.X^T=b+\sum_{i=1}^{n}w_ix_i \qquad (3)$$

The output O produced by the perceptron is no longer determined by a threshold value but by a transfer function f. In most cases the transfer function is of the log-sigmoid or the tan-sigmoid forms depicted below :

$$O=f(N)=\frac{1}{1+e^{-N}} \qquad (4)$$
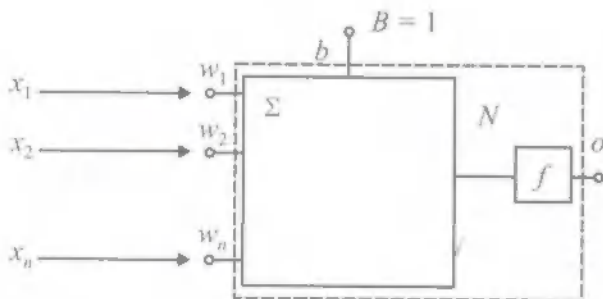
$$O=f(N)=\frac{1-e^{-N}}{1+e^{-N}}$$

Fig 3. The perceptron

To solve a problem requires two steps : a training phase and a testing phase. During the training phase, a set of inputs are fed to the neural unit and the outputs they should produce are also known, and these are called targets. For example, for simulating an AND gate an input of [1, 0] should produce a target of [0] and an input of [1, 1] should produce a target of [1]. The weights are not known, so initial estimates are assumed (often random values or all zeroes). The actual output O produced can be calculated from Eq. (3) and Eq. (4) above. If the output does not match the target then an error is produced and this error is used to modify the

weights in such a way that subsequent errors are reduced. This constitutes an iteration and is called an epoch. In the next iteration the inputs are again fed to the unit and the new weights are used to calculate the error. This process is repeated iteratively until the errors are all reduced to zero or some pre-defined small value. The unit is said to have converged and the final weights are called the balanced weights. The balanced weights provide a representation of the problem pattern since for all inputs these weights produce the correct outputs.

During the *testing* phase, an unknown set of inputs are fed to the neural unit and the balanced weights are then used to calculate the correct outputs. The power and flexibility of neural units lie in the fact that the test inputs need not be exactly identical with any of the training set inputs but only similar, for the perceptron to produce the correct decision. This property is frequently used to solve problem like pattern recognition and character recognition, where training is done using a separate set of characters and testing is done on another set of similar characters e.g. produced using a different font.

In most real-world problems instead of using a single perceptron, we use a network of connected neural units since we want multiple outputs at the same time. Such networks often have multiple layers of neural units between the input and output layers, in which case it is called multi-layered perceptrons (MLP) and the in-between layers are called hidden layers. See Fig. 4.
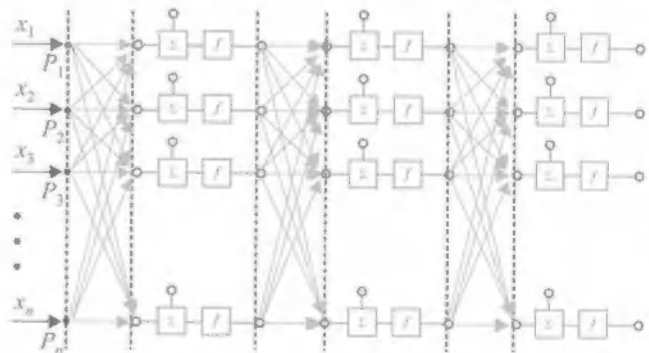

Fig. 4. A neural network

In MLP computations multiple errors are produced at the output corresponding to each neural unit, so a cumulative error called the Mean Square Error (MSE) is used for updating the weights, as depicted below, where ei is the error produced at the i-th output unit and there are m such units at the output

$$MSE = \frac{1}{m}\sum_{i=1}^{m}e_i^2 \qquad (5)$$

4. Experimentations and Results

4.1 Dataset

Skin texture images downloaded from Dermnet picture collection (http://www.dermnet.com/) are used in the experiments. The dataset consists of a total of 180 images divided into three disease classes: Acne (Class-A), Ichthyosis (Class- I) and Keratosis (Class-K) with 60 images per class. Each image is 128 by 128 pixels in dimension and in GIF file format. A total of 90 images are used as the Training set (T) and the remaining 90 images as the Testing set (S). The images are arranged in the order A, I, K i.e. the first subset of 30 images of training (or testing) set belongs to Class-A, the next subset to Class-I and the last subset to Class-K. Sample images of classes A, I, K are shown in Figure 5.

For computing recognition rates, features are first considered individually, then in two-dimensional (2-D) feature space and finally in three-dimensional (3-D) feature spaces, involving multiple GLCM features simultaneously. Comparisons between training and testing sets are done using multi-layer perceptrons. At each stage the best results are tabulated and the corresponding discrimination plots are reproduced. The legends used in this work are listed in Table 1. Here X denotes a class name which can be either A or I or K.

(a)

(b)

(c)

Fig. 5. Samples of medical images belonging to three skin condition classes : (a) A (b) I © K

Table 1 : Legends

|	
TXC	Training set, Class-X, GLCM Contrast
TXH	Training set, Class-X, GLCM Homogeneity
TXM	Training set, Class-X, GLCM Mean
TXV	Training set, Class-X, GLCM Variance
TXN	Training set, Class-X, GLCM Energy
SXC	Testing set, Class-X, GLCM Contrast
SXH	Testing set, Class-X, GLCM Homogeneity
SXM	Testing set, Class-X, GLCM Mean
SXV	Testing set, Class-X, GLCM Variance
SXN	Testing set, Class-X, GLCM Energy
X	Class name, either A or I or K

4.2 Individual Features

Values of individual GLCM features C, H, M, N, V defined in Eq. (2) for training and testing images for each of the three classes are computed. Test images are compared to training clusters using NN classifiers. Accuracy results are summarized in Table 2. The first column depicts the feature used, the second column shows the neural network configuration (NNC) viz. 1-10-3 indicates 1 input unit (for the individual feature), 10 units in the hidden layer and 3 units in the output layer (corresponding to the 3 classes to be distinguished). The third, fourth and fifth columns indicate the percentage recognition accuracies for the three classes, the sixth column provides the overall accuracy for the three classes and the last column indicates the best Mean Square Error (MSE) obtained during the training phase of the NNs. In all cases the NN classifiers were run for 50000 epochs. Feature values were appropriately scaled to lie within the range 0 to 1 before being fed to the classifier.

Table 2 : Accuracy results for single features

F	NNC	A	I	K	O	MSE
C	1-10-3	40	90	36.6	55.5	0.16
H	1-10-3	23.3	66.6	40	43.3	0.17
M	1-10-3	86.6	93.3	46.6	75.5	0.10
N	1-10-3	53.3	53.3	30	45.5	0.17
V	1-10-3	43.3	0	40	27.8	0.20
Avg		49.3	60.6	38.6	49.5	

From Table 2 it is observed, that best results are produced by M (75.5%) and C (55.5%). Corresponding plots depicting the variation of these feature values for the three classes over the training and testing datasets are shown below.
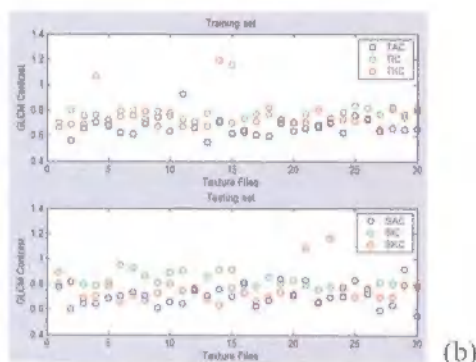

(a)
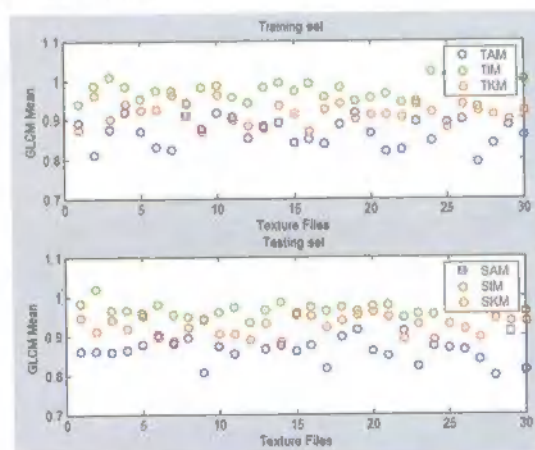
Fig. 6. Feature plots for (a)TAM, TIM, TKM, SAM, SIM, SKM (b) TAC, TIC, TKC, SAC, SIC, SKC

Fig. 7. Feature plots for (a) TM vs. TN, SM vs. SN (b) TM vs. TH, SM vs. SH

4.3 Joint Features in 2-D Feature Space

To improve upon the results obtained using individual features, joint features are next considered in two-dimensional feature spaces i.e. C-H, C-M, C-N, C-V, H-M, H-N, H-V, M-N, M-V, N-V. Accuracy results are summarized in Table 3. In all cases the NN classifiers were run for 50000 epochs. Feature values were appropriately scaled to lie within the range 0 to 1.

4.4 Joint Features in 3-D Feature Space

To improve upon the results obtained using individual features, joint features are next considered in three-dimensional feature spaces i.e. C-H-M, C-H-V, C-H-N, C-M-V, C-M-N, C-N-V, H-M-V, H-M-N, H-N-V, M-N-V. Accuracy results are summarized in Table 4. In all cases the NN classifiers were run for 50000 epochs. Feature values were appropriately scaled to lie within the range 0 to 1.

Table 3 : Accuracy results for joint 2D features

F	NNC	A	I	K	O	MSE
C-H	2-100-3	63.3	73.3	33.3	56.6	0.09
C-M	2-100-3	76.6	50	53.3	60	0.07
C-V	2-100-3	43.3	56.6	46.6	48.9	0.10
C-N	2-100-3	66.6	50	43.3	53.3	0.09
H-M	2-100-3	76.6	93.3	63.3	77.8	0.05
H-N	2-100-3	20	36.6	43.3	33.3	0.08
H-V	2-100-3	36.6	30	40	35.5	0.13
M-N	2-100-3	83.3	93.3	63.3	80	0.06
M-V	2-100-3	73.3	40	63.3	58.9	0.06
N-V	2-100-3	43.3	43.3	40	42.2	0.12
Avg		58.3	56.6	48.9	54.6	

Table 4 : Accuracy results for joint 3D Features

F	NNC	A	I	K	O	MSE
C-H-M	3-150-3	73.3	86.6	63.3	74.4	0.04
C-H-M	3-150-3	56.6	30	36.6	41.1	0.11
C-H-M	3-150-3	53.3	30	66.6	50	0.06
C-H-M	3-150-3	66.6	26.6	40	44.4	0.07
C-H-M	3-150-3	86.6	93.3	66.6	82.2	0.04
C-H-M	3-150-3	53.3	50	46.6	50	0.09
C-H-M	3-150-3	83.3	80	63.3	75.5	0.04
C-H-M	3-150-3	76.6	90	53.3	73.3	0.05
C-H-M	3-150-3	60	33.3	53.3	48.9	0.08
C-H-M	3-150-3	93.3	63.3	60	72.2	0.04
C-H-M		70.3	58.3	54.9	61.2	

From Table 3 it is observed that best results are produced by M-N (80%) and H-M (77.8%). Corresponding plots depicting the variation of these feature values for the three classes over the training and testing datasets are shown below.
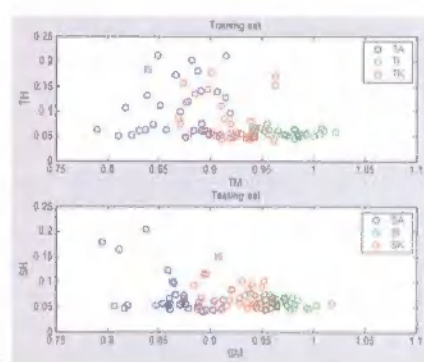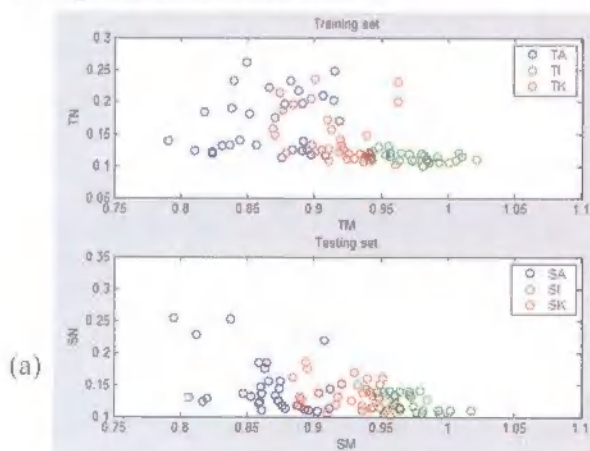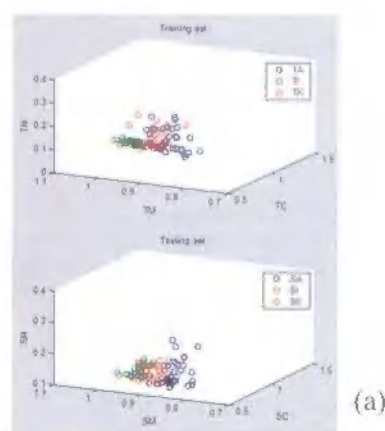
From Table 4 it is observed that best results are produced by C-M-N (82.2%) and H-M-V (75.5%). Corresponding plots depicting the variation of these feature values for the three classes over the training and testing datasets are shown below.
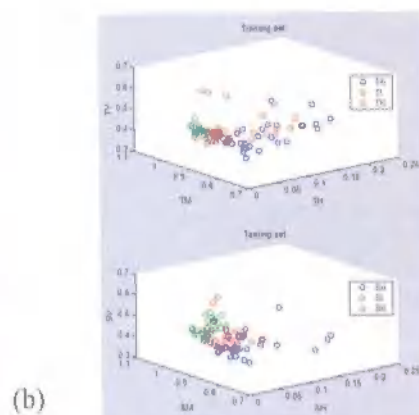

(a)

(a)

Fig. 8. Feature plots for (a) TC vs.TM vs. TN, SC vs. SM vs. SN (b) TH vs. TM vs. TV, SH vs. SM vs. SV

5. Analysis

Automated discrimination between three skin texture classes was done using a variety of approaches to find the optimum results. Out of five GLCM based features considered individually, M produced the best recognition rate of 75.5%. Among joint 2-D feature spaces M-N produced the best result of 80%. Joint 3-D feature spaces were seen to improve on the accuracy rates to 82.2% using C-M-N. Best results are summarized below.

Table 5 : Best performance results

	Individual	Joint 2D	Joint 3D
F	M	M-N	C-M-N
%	75.5	80.0	82.2

Out of the disease classes, I was the best recognized using single features (60.6%), while A was the best recognized using 2-D features (58.3%) and 3-D features (70.3%).

The best recognition result, by using a combination of Contrast, Mean, Energy features, was obtained by using a neural network having a configuration of 3-150-3 i.e. 3 input units, 150 units in the hidden layer and 3 output units. The three feature vectors C, M, N of each training sample, were fed to the three inputs of the neural net and it was trained in this manner by a total of 90 samples, 30 per class. The perceptron took 50000 epochs to converge to an MSE value of 0.04. The network convergence and output plots are shown below. The output plot depicts a 86.6% accuracy for recognizing disease A, 93.3% for B and 66.6% for C, each class being represented by 30 test samples.
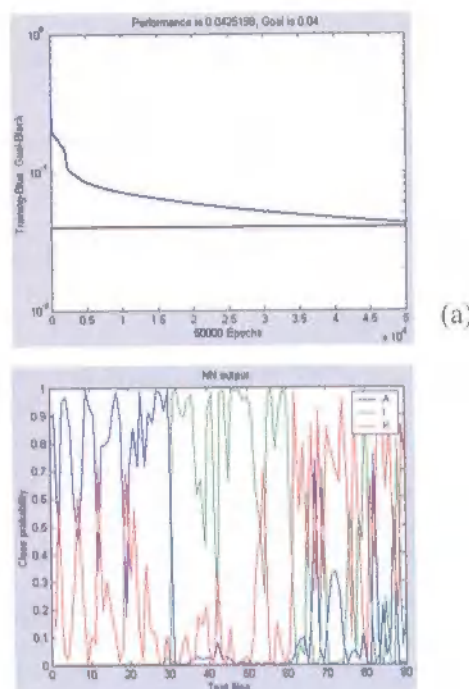

Fig. 9. NN convergence and output plots for the C-M-N feature vector

To put the above results in perspective with the state-of-the-art, the best results reported in [Shyu, 1999] for identifying disease classes from 302 lung section images involving texture features homogeneity, contrast, correlation and cluster, in addition to other features like grey-level histogram, is 76.3%. Accuracy for classification of 800 endoscopic images in [Xia, 2005] using a fusion of color, texture and shape features ranges from 77% to 90% but only about 25% involving texture features alone. Accuracy results reported in [Alabbadi, 2008] tested on 300 skin texture images is 96% but uses 9 color features in addition to 4 texture features, entropy, energy, contrast, homogeneity.

6. Conclusions and Future Scopes

This paper proposes an automated system for recognizing disease conditions of human skin by analyzing skin texture images using texture recognition techniques. Skin disease conditions differ in appearance in a way which cannot be modeled appropriately by specific colors and can best be identified using statistical variation of texture. Such automated medical diagnosis systems can prove extremely useful where there might be a dearth of good medical professionals. On one hand this would be useful for dermatologists to reduce diagnostic errors, while on the other it can serve as the initial test bed for patients before seeking expert advice.

The study reveals that performance improved when joint features were considered as compared to individual features. A salient feature of this approach is the low-complexity data modeling scheme whereby a small number of scalar values are used to represent image content, instead of multi-dimensional vectors like histograms. This makes the system low on computational overheads, and makes it suitable for use in remote and rural sectors, where computational resources can be scarce. Low resources also imply low cost involvements.

The accuracy of the current system is comparable to those reported in contemporary works. Most of the other works have dealt with 24-bit images and have utilized color based features in addition to texture based features. In comparison the current work has used 8-bit images and only texture based features. It is expected that accuracy results can be improved upon by using: (1) Color features along with texture, by employing GLCMs on individual R, G, B color channels of 24-bit images (2) Normalization of the brightness and contrast of the images by pre-processing, involving histogram equalization, before calculation of features.

References

1. H. Muller, N. Michoux, D. Bandon, A. Geissbuhler, "A review of content-based image retrieval systems in medical applications – clinical benefits and future directions," *Int. J. of Medical Informatics*, vol. 73, 2004, pp. 1-23.
2. D. W. Bates, N. Spell, D. J. Cullen, E. Burdick, N. Laird, L. A. Peterson, S. D. Small, B. J. Sweitzer, L. L. Leape, "The costs of adverse drug events in hospitalized patients", *Journal of the American Medical Association*, vol. 277, no. 4, 1997, pp. 307-311.
3. N. S. Weingart, R. M. Wilson, R. W. Gibberd, B. Harrison, "Epidemilogy of medical error", *BMJ*, vol. 320, 2000, pp. 774-777.
4. D. Copec, M. H. Kabir, D. Reinharth, O. Rothschild, J. A. Castiglione., "Human errors in medical practice : systematic classification and reduction with automated information systems," *Journal of Medical Systems*, vol. 27, no. 4, 2003, pp. 297-313.
5. R. Wang, A R Hanson, E M Riseman, "Texture analysis based on local standard deviation of intensity", *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, Florida, 1986, pp. 482-488,
6. H. Tamura, S. Mori, T. Yamawaki, "Textural Features corresponding to visual perceptions", *IEEE Transaction on Systems, Man and Cybernetics*, vol. 8, no. 6, 1978, pp. 460-473.
7. A. P. Pentland, "Fractal based description of natural scenes", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, 1984, pp. 661-674.
8. N. K. Huang, "Markov model for image segmentation", *Proc. 22nd Allerton Conf. on Communication, Control and Computing*, Montecello, 1984, pp. 775-781.
9. A. C. Bovik, M. Clark and W. S. Geisler, "Multichannel texture analysis using localized spatial filters", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, 1990, pp. 55-73.
10. N. K. Al abbadi,. N. S. Dahir, Z. A. Alkareem., "Skin texture recognition using neural networks," *Proc. Int. Arab Conf. on Information Technology*, Tunisia, 2008.
11. P. Rubegni et al., "Automated diagnosis of pigmented skin lesions", *Int. Journal on Cancer*, vol. 101, 2002, pp. 576-580.
12. J. M. Blackledge, D. A. Dubovitskiy, "Texture classification using fractal geometry for the diagnosis of skin cancers", *EG UK Theory and Practice of Computer graphics*, W. Tang, J. Collomosse (eds.), 2009, pp. 1-8.
13. S. Shahreza, M. Mousavi, "A new Bayesian classifier for skin detection", *Proc. 3rd Int. Conf. on Innovative Computing, Information and Control*, 2008, pp. 172-175.
14. R. M. Haralick, "Statistical and structural approaches to Texture", *Proc. IEEE*, vol. 67, 1979, pp. 786 – 804.
15. C. R. Shyu, A. Kak, A. Kosaka, "ASSERT a physician in the loop CBRS for HRCT image databases". Comp. Vision and Image Understanding, vol. 75, no. 1, 1999, pp. 111-132.
16. S. Xia, W. Mo, Z. Zhang, "A content based retrieval system for endoscopic images", Int. Journal of Information Technology, vol. 11, no. 2, 2005, pp. 27-32.

Dr. Ranjan Parekh is a faculty at the School of Education Technology, Jadavpur University, Kolkata. His research interests include multimedia databases and medical imaging. He is the author of the book "Principles of Multimedia" published by McGraw-Hill.
ranjan_parekh@yahoo.com